Lawrence Livermore National Laboratory

Data Management of Metadata-Rich File Systems

Sasha Ames

Maya Gokhale (LLNL), Carlos Maltzahn (UCSC)

LLNL-PRES-418053

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Introduction

- 58% annual increase of data creation rate, 5-fold increase in 4 years. 487 exabytes created in 2008 [Gantz08,09]
- Hierarchical file systems [Daley65] originated in 1965 for 1000s of files.
  - Now 6 or more orders of magnitude more files per file system
  - Names and hierarchical directories as the only user-defined metadata concepts not adequate anymore

➡ Need scalable data model for organizing files

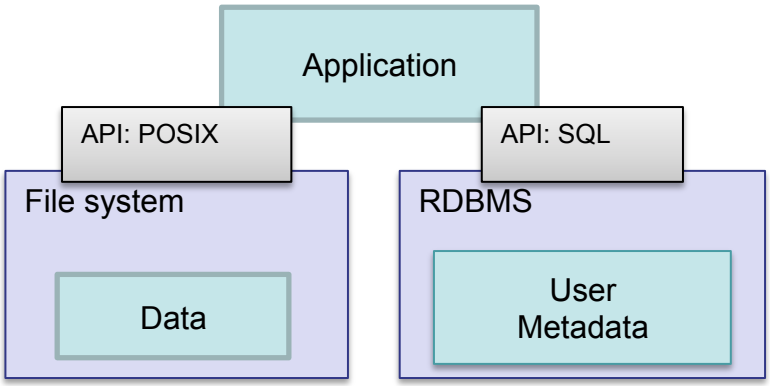
➡ Need scalable naming interface for accessing files

➡ Example: find files of “news documents” that refer to the “location” “New York”

Traditional Architecture

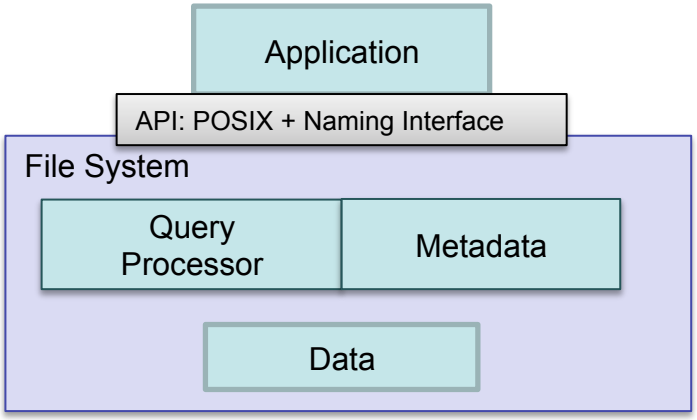
- Relational databases store metadata
- File systems store data
- Application must "bridge" two systems
- Advantages:
  - High throughput for file I/O
  - Mature technology for metadata management and retrieval

- Problems:
  - Disparate name spaces
  - Brittle schema
  - Brittle consistency
  - Individual file stat/open/close



Metadata-rich File Systems

- Integrates database functionality into file system for the purpose of metadata management
- Benefits:
  - General schema
  - Unified namespace
  - Robust consistency
  - Query based file access interfaces

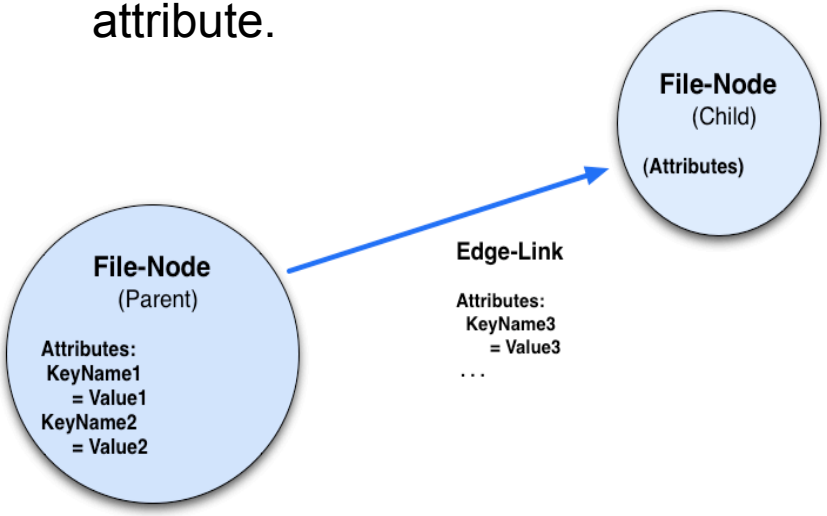


- Our approach:
  - Graph-based data model for file metadata
  - Path-based language interface for query added to POSIX file system interface
    - Query results appear in directory listings (Semantic File Systems)

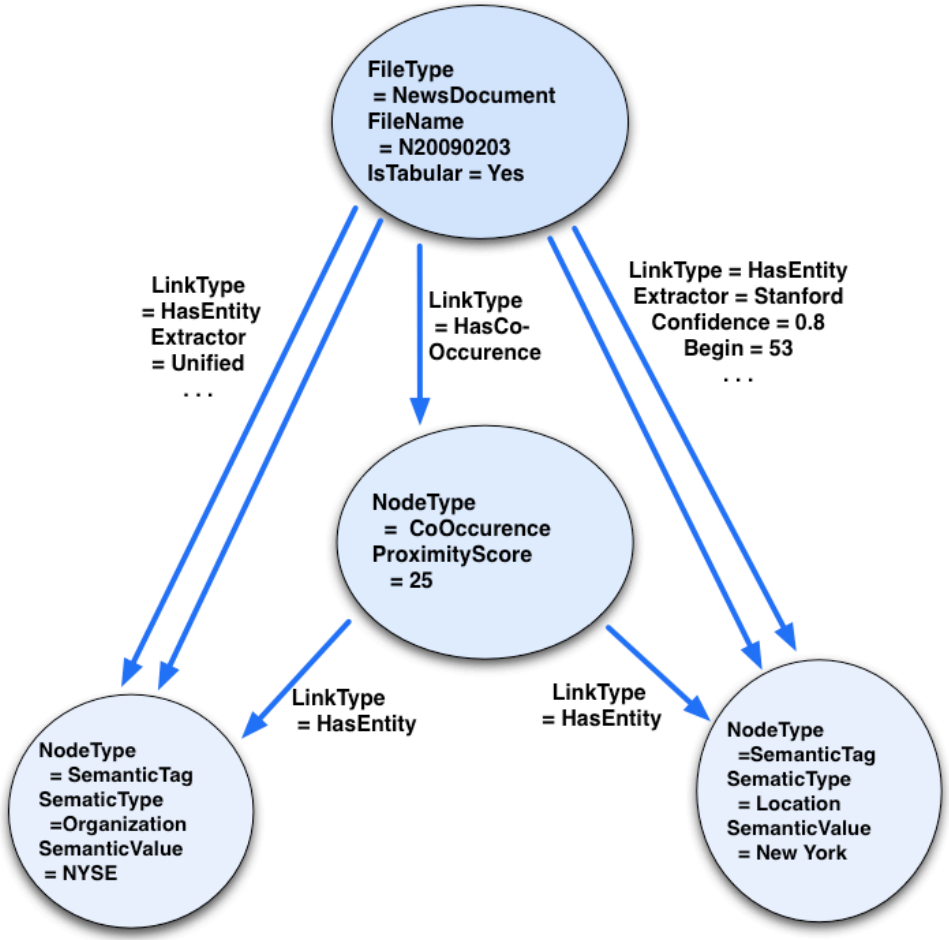
Data Model

- Unlike POSIX, no file/directory distinctions
- Files connected with directed "edge-links"
- Name=value paired attributes attached to files and links.
- File and links assigned IDs
- File "names" become one of many attributes
- Links identified through file endpoints and/or attributes
- POSIX directories are zero-byte files linking to children (for semantics compliance)

Use of zero-byte files helpful in attaching to files metadata that is more complex than a single attribute.



Example graph data

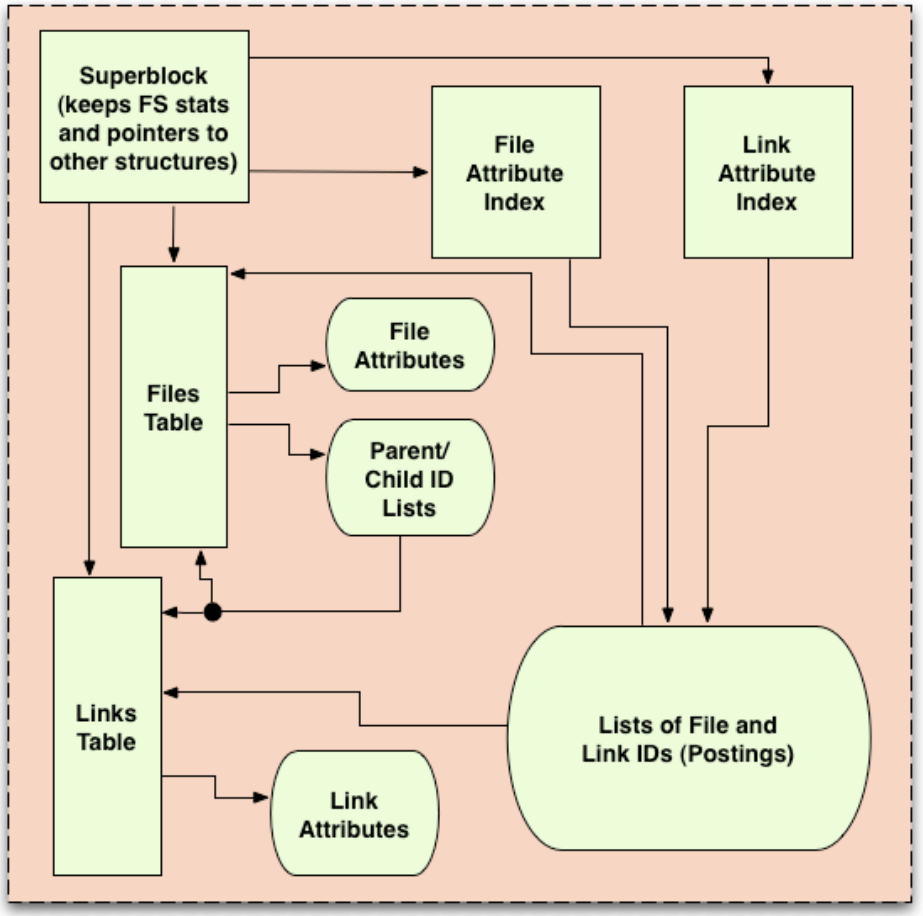


Query: find files of “news documents” that refer to the “location” “New York”

Quasar expression:
@FileType=NewsDocument
@child:SemanticType=Location;
SemanticValue=New York;
^Extractor=Stanford

QFS Physical Data Model

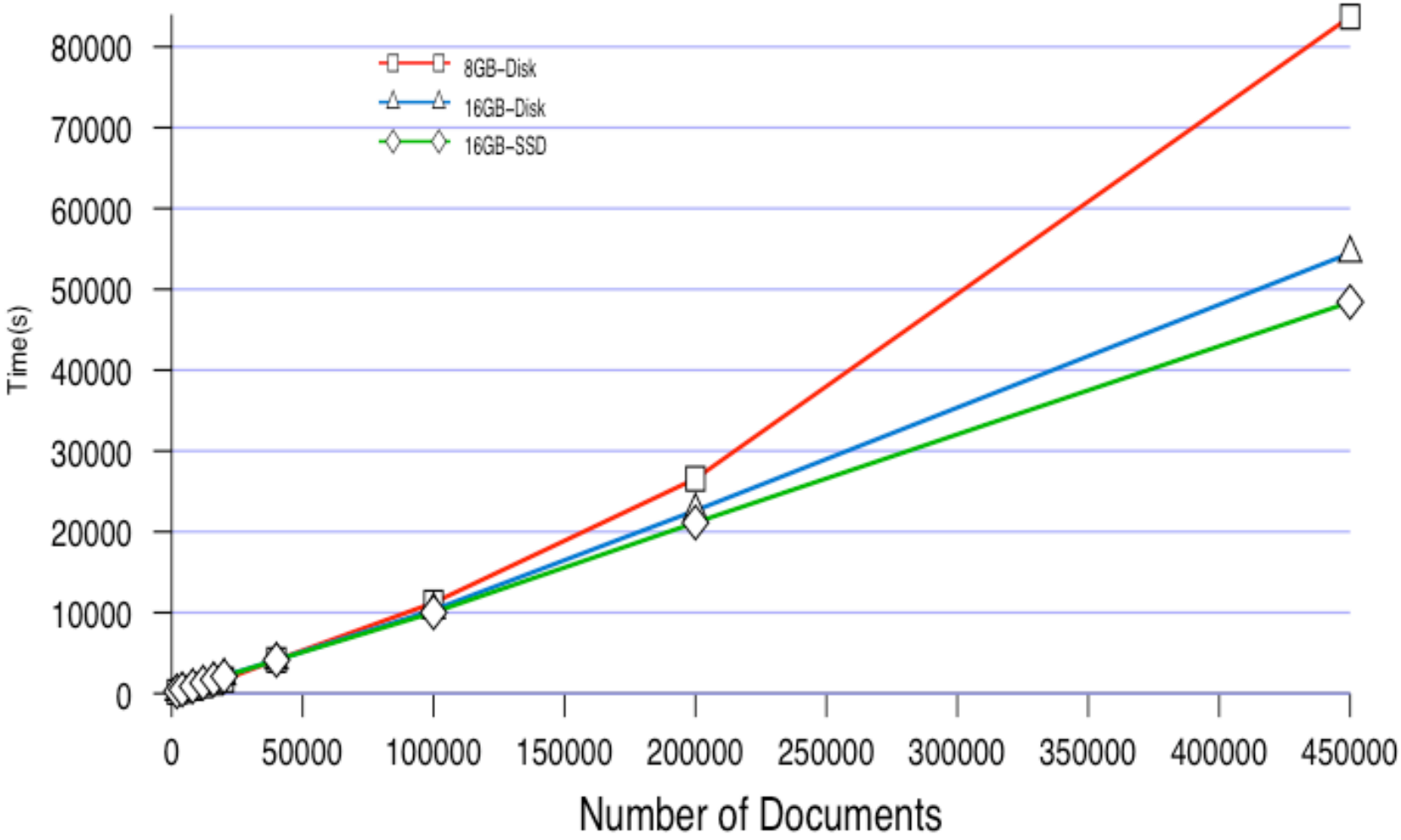
- Table entries refer to file attribute sets and lists of ID pairs, each containing parent/child file ID (inode #) and link ID
- Link ID refers to links table entries
- File IDs refer back to file table entries
- Global file/link indices contains attribute vocabulary trees
- Each index tree node refers to a posting list of file/link IDs



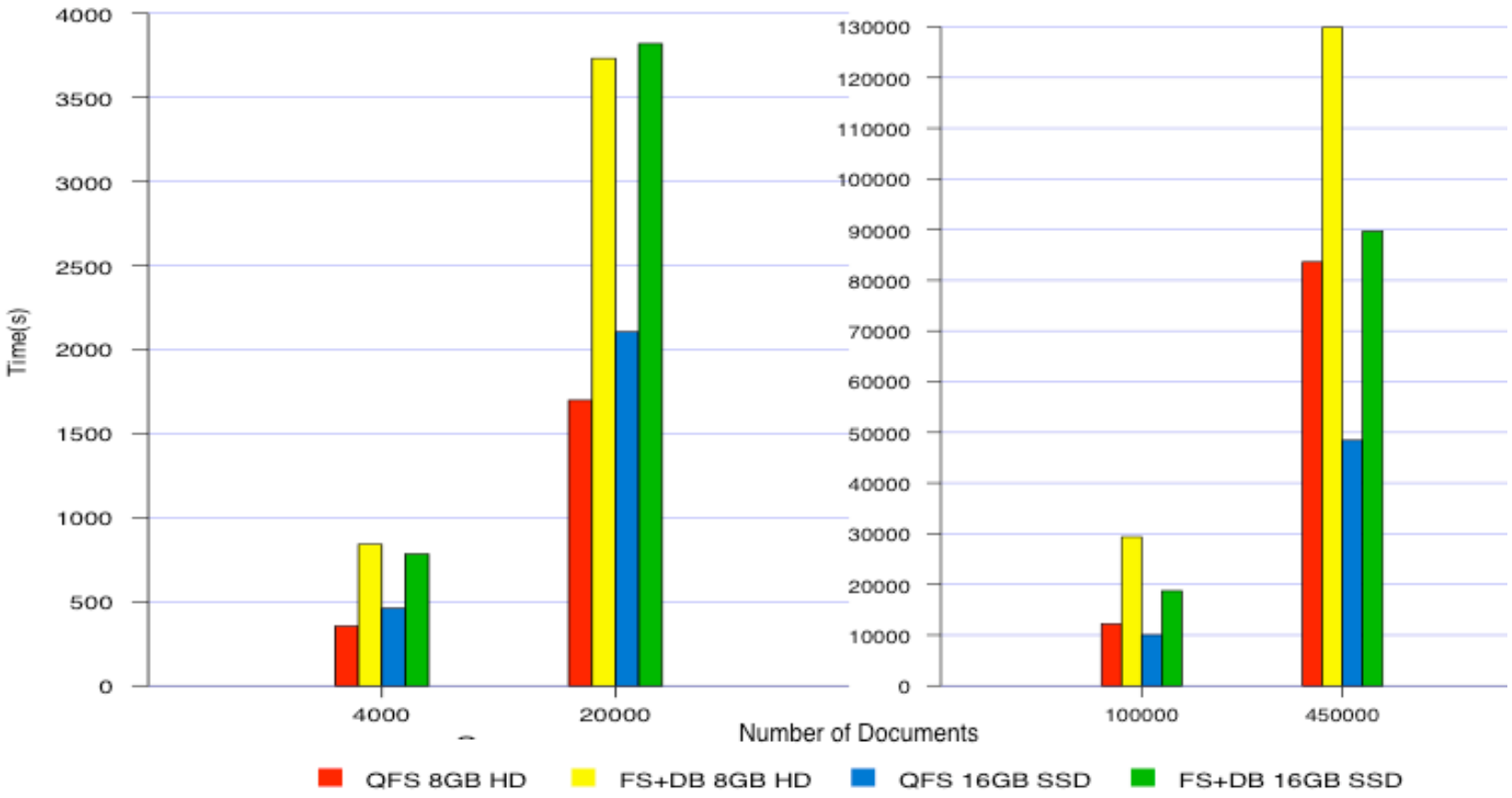
Experimental Methodology

- Workload Studies
  - Use Livermore Entity Extractor for data + metadata, Reuters news corpus
    - Extractor reconfigured for QFS
  - Compare QFS prototype with FS + DB (PostgreSQL)
    - DB configured with schema specific to Lextrac and indexing on all columns
  - Ingest experiments: Look at increasing document counts, 8/16GB, HD/ SSD.
  - Query experiments: 5 classes of queries.
    - Choose query “terms” from Lextrac entity distributions.
    - Equivalent queries run in both Quasar and SQL
- POSIX FS operation studies: microbenchmarks to measure overhead of QFS metadata management
  - Create a directory tree (mkdir)
  - “Find” (stat/opendir/readdir)
  - Move single file (rename)

Ingest Study 1 – QFS scalability



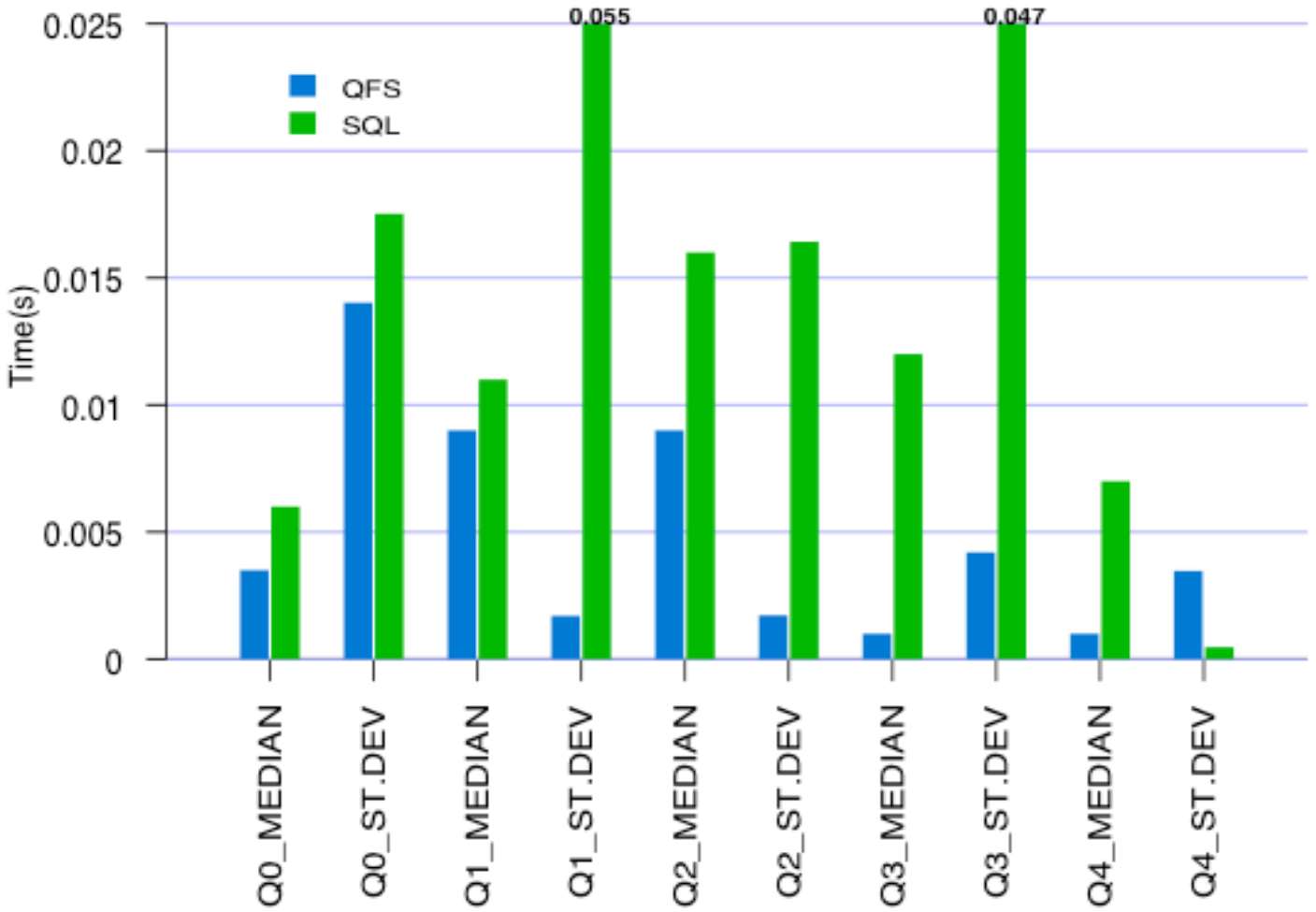
Ingest Study 2 – QFS vs FS+DB



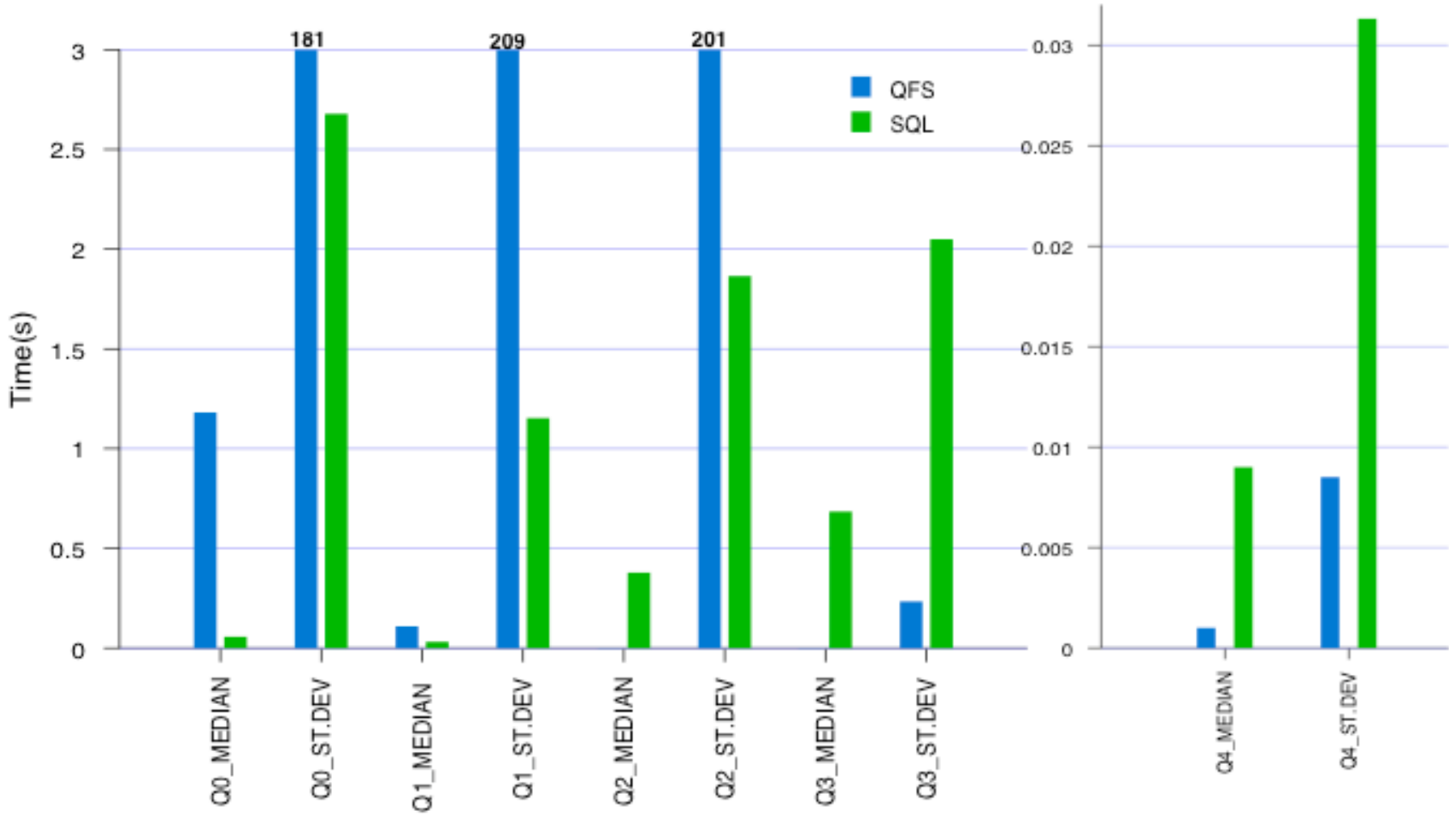
Example Queries

- **Simple Queries:**
  - **Q0** Find all documents containing the "location" “New York.”
  - Terms based on ranking entity values by frequency
- **Complex Queries:** (Document, entity1, entity2, proximity)
  - **Q1** Find all documents that contain “New York” and “NYSE” with proximity score of “25”.
  - **Q2** Find the proximity scores relating “New York” and “NYSE” in documents with names in the range of “N20090101” – "N20090331.”
  - **Q3** Find entities co-occurring with “New York” in documents with names in the range “N20090101” – “N20090331” whose proximity score with “New York” is between “20” and “30”.
  - **Q4** Same as Q3 but match scores of exactly "25".
  - Entity values, proximity scores and document ranges selected randomly (chance of no results)

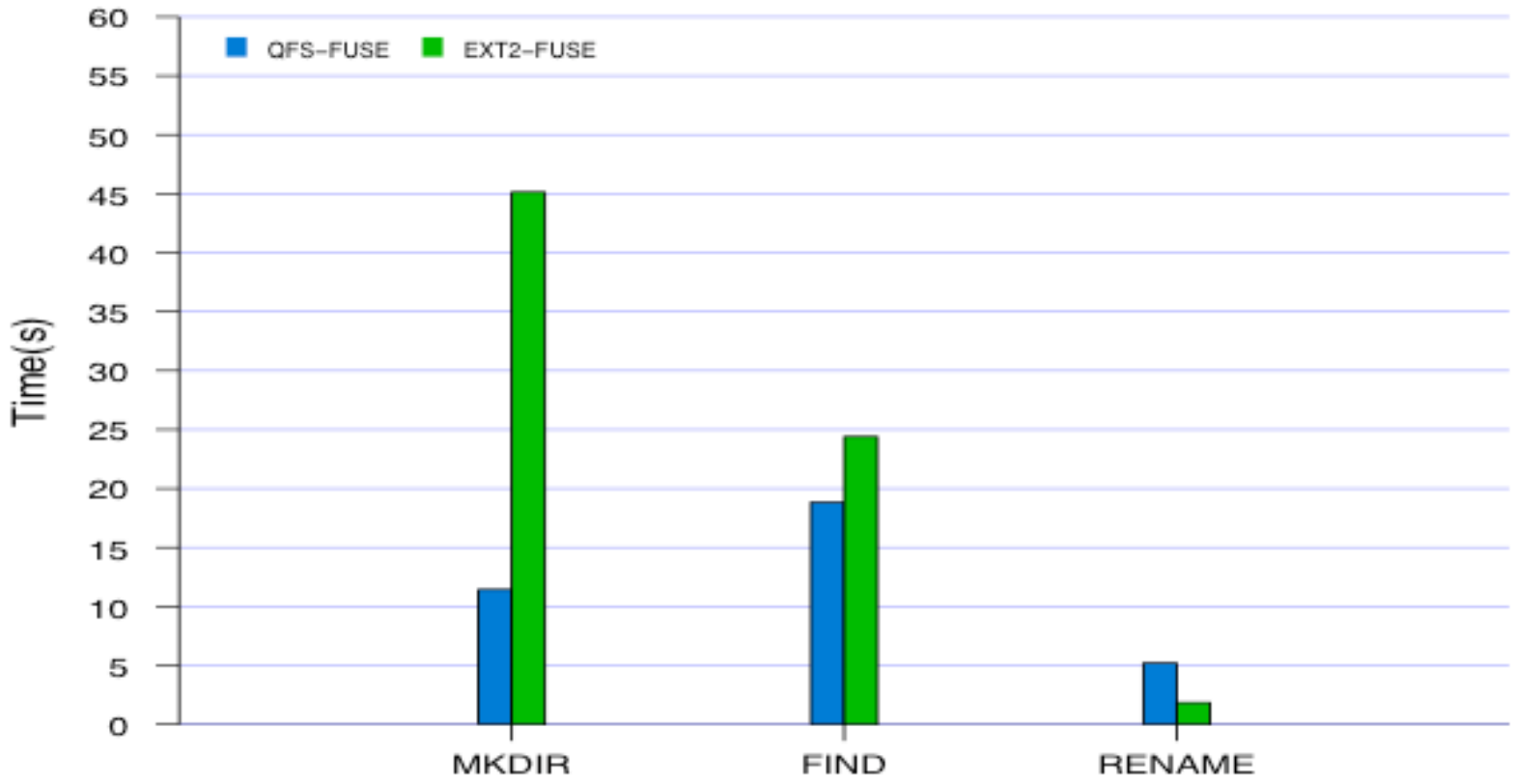
Query Study 1 – 20,000 Lextrac Documents



Query Study 2 – 450,000 Lextrac Documents



POSIX Operation Microbenchmarks



Discussion and Future Work

- QFS prototype optimized for ingest
  - Ingest 1.4-1.85x faster than FS+DB
  - File moves slower than ext2 FS
- QFS Metadata store 3.5x larger than Postgres
  - 450K documents Q0, 20% of Q1-Q2 have 1000x slower performance
- Future Work:
  - Scientific data use case: Sloan Digital Sky Survey
  - Metadata analysis: look at index properties and access patterns
  - Optimizations
    - Reduce metadata store size
    - Improve query planner